

协作编程中的计算思维发展轨迹研究*

——基于量化民族志的分析方法

□吴忞 王戈

摘要：伴随科学技术的迅猛发展，计算思维能力成为当前社会各行业人才都应具备的重要技能。计算机教育领域的研究发现，协作编程能够更好地帮助学生进行编程学习，但是在协作过程中，学生计算思维的动态发展情况却难以测评。“量化民族志”（Quantitative Ethnography）既是一种对质性数据进行量化分析的方法，也是一种对专业能力进行网络化建模的技术，为协作学习过程的分析和评价提供了新的思路。将该方法应用于分析协作编程活动过程中小组成员之间的会话交流内容，有利于构建学生计算思维能力不同维度之间关联的认知网络模型，并揭示不同能力水平的小组在计算思维能力水平上的差异和发展轨迹。以本科一年级的“C/C++程序设计基础”课作为研究情境开展的实证研究发现，高水平组和低水平组学生的计算思维网络结构在初期虽有显著不同，但是随着协作任务的不断深入，两组学生的思维模式趋于相似。该研究结果表明协作编程能够促进初学者计算思维能力的提升，且量化民族志方法能为计算思维的能力评估提供一种基于质性数据的大规模测评方法。

关键词：计算思维；协作编程；量化民族志；认知网络分析

中图分类号：G434 **文献标识码：**A **文章编号：**1009-5195(2019)02-0076-10 doi:10.3969/j.issn.1009-5195.2019.02.009

***基金项目：**教育部人文社会科学青年基金“协作问题解决能力在线测评研究”（16YJC880085）；华东师范大学教学改革与研究项目（40400-10201-511232/115）。

作者简介：吴忞，博士，副教授，硕士生导师，华东师范大学教育信息技术学系；王戈，硕士研究生，华东师范大学教育信息技术学系（上海 200062）。

一、引言

近年来，移动通讯、普适计算、物联网、云计算、大数据等新概念和新技术的出现，在社会各方面引发了一系列革命性的突破，极大地改变了人们对计算和计算机的认识。庞大的数据资源使得学术界、商界、政府等各个领域开始其数字化进程。随着这一进程的不断深入，无处不在的计算思维成为人们认识和解决问题的能力之一。2016年，我国高等教育教学指导委员会推动和组织了有关在大学计算机课程体系中融入“计算思维”培养的相关研究工作。在计算思维的教学体系中，应正确处理好知识、能力和思维的关系，通过讲授计算机的基本理论和技术，揭示计算思维的内容。但是由于计算机语言语法逻辑严谨，初学者难以熟练使用计

算思维解决问题，且课堂教学任务繁重，学生学习编程的自信心不足，学习兴趣也大大下降。采用何种教学策略解决普遍性问题是研究人员所关注的核心。已有研究表明，团队协作是一种有效的教学策略。它能够模拟软件公司中实际的软件开发工作模式，带给学生真实的软件开发项目体验，提高学生的协作能力和编程技能。在项目实践中，学习者也可相互学习，从而提升编程的自信心和学习兴趣，达到共同进步的目的。通过让学生以团队协作的形式完成编程项目的教学方法又被称为“协作编程”。协作编程作为一种计算机教育方式有诸多优势，比如，提高编程学习效率，减少编程错误，促进相互学习，提高问题解决的自我效能等（Williams & Kessler, 2002; Denner et al., 2014）。计算机教育中越来越多的协作编程教学方法的应用表明，协作

编程已成为计算机编程教育的发展趋势。

目前关注协作编程的研究人员主要关注对学生编程结果的评价,如学生的课程成绩分析、编程项目成绩分析、编程代码质量分析等。但团队协作完成的作品并不能完全反映小组中每一位学生参与协作编程时的投入程度、思维过程和能力水平。Lye和Koh(2014)指出可以通过记录学生在编程时口头表达的思维过程(又称有声思维),并结合其编程过程进行分析。然而,学生在参与讨论时的话语会涉及计算思维的不同能力维度和要素,如何动态建模学生计算思维各维度之间的发展变化及其相互关系是质性分析的难点。其次,学生的编程能力水平不同,不同小组的协作编程行为模式也各异,这对探索不同编程学习阶段的协作编程行为模式,及其所反映的编程团队的计算思维能力发展情况提出了挑战。因此,本研究通过分析游戏开发任务中初学者的协作编程过程和完成编程任务的策略方式,建立计算思维发展轨迹模型,进而为培养计算思维能力的协作编程学习模式提供实证依据。

二、理论基础

1. 计算思维

“计算思维”一词最早由美国麻省理工学院的西蒙·帕佩特(Seymour Papert)教授提出,美国卡内基梅隆大学计算机系主任周以真教授对其进行了进一步的阐释,指出计算思维不仅仅是计算机科学家应具备的能力,它是每一个人都应学习和熟练运用的专业思维方式和技能。然而,对计算思维概念的界定学术界仍存在不少争议。例如,Chao(2016)将计算思维视作问题解决的过程,并从计算设计(理解问题、设计解决方案),计算实践(解决问题)和计算表现(测试方案)三个方面对计算思维进行研究。该分类侧重于关注计算思维问题解决的认知水平。也有学者将计算思维视为一种更为全面的能力,如Brennan和Resnick(2012)认为,计算思维包括计算概念(核心概念)、计算实践(应用概念)和计算认识(对计算思维的作用和局限的认识)。虽然学者们对于计算思维概念的界定没有达成共识,但大家普遍认为,计算思维并不是单一的认知技能,而是不同能力维度的有机结合。因此,在研究和评

估计算思维能力时,应该考虑到计算思维不同维度的层次结构,以及各维度的子能力和不同子能力之间的联系。

计算思维提出后受到众多计算机教育领域研究者的关注。已有研究表明,关于培养计算思维能力的课程设计中,计算机编程教育是最常见的(Lye & Koh, 2014)。编程不仅能培养学生的计算思维能力,也能作为计算思维能力的示范(Grover & Pea, 2013)。以往关于编程教育的研究往往侧重于对初学者编程能力的考察,缺乏对计算思维能力在编程学习阶段发展情况的全面评估。此外,计算思维能力的评估通常侧重于总结性评价,如问卷和测验分数(Zhong et al., 2016; Korkmaz et al., 2017; Román-González et al., 2017),而对编程促进计算思维发展的形成性评价较少,且传统的评价更侧重于单一的编程结果。例如,Kafai等人(2014)依据Brennan和Resnick的三维框架,研究了学生在三个维度上计算思维的发展情况,结果发现编程代码的混合重组对计算思维能力的发展有显著作用。但是,该研究并没有考虑计算思维三个维度之间的联系,而是分别对每个维度进行了评估。在此基础上,Chao(2016)探索了三者之间的关系,并通过聚类分析识别出编程学习的独特模式。但他对这些关系的描述仍然是粗粒度的,而且也没有将计算思维视作整体来考察各维度间的协同发展。

2. 协作编程

协作编程(Cooperative Programming)被定义为一组学生通过分别扮演代码编写、任务分析和代码评估等不同角色,在交流讨论过程中,共同完成编程任务、解决编程问题的教学方法。这种教学方法源于认知和社会建构主义的教学理念,将社会性学习视为一种基于先备知识和社交互动体验的积极建构新知的过程(Kalaian & Kasim, 2014)。它也反映了认知要素可以分布在不同的人、工具和作品之中的观念(Berland & Lee, 2011; Mangalaraj et al., 2014)。协作编程一直是研究者认为能够支持项目编程的有效方法,尤其是对经验不足的学习者来说,该方法更有助于培养计算思维和掌握编程知识(Denner et al., 2014),同时也有助于提升学习者的问题解决能力和编程学习的自信心(Beck & Chizhik, 2013)。但

是,目前的研究仍然缺少在皮亚杰认知发展观点基础上,对传统的新手程序员编程特征模式的拓展,比如对团队协作如何帮助协调组内角色,建立共识,解决真实编程任务的研究。

在以往关于计算机编程教育的实证研究中,研究者往往面临两个困境。首先,对编程代码质量的分析仅局限于对计算思维的基本知识和技能的测量(Wang & Hwang, 2012),学生在编程作品中出现的代码元素并不一定能说明其对编程问题本质的理解(Brennan & Resnick, 2012)。为了更好地研究包括计算实践和计算认识在内的深度学习,可以通过记录学生在编程时口头表达的思维过程,捕捉和分析他们的屏幕编程过程(Lye & Koh, 2014)。其次,研究者们往往将不同的维度割裂分析,而没有将他们视为相互关联、协同发展的能力(Khosa & Volet, 2014; Kwon et al., 2014)。在Shaffer (2006)看来,专业知识、技能、学科认识论、价值观和专业身份等是相互交织和促进的,专业能力正是通过这些不同能力维度之间的联系来体现。例如,在某个专业领域(如医学)塑造一个人的个人身份(如医生)需要通过实践(如治病)来展示他在该领域的能力,反过来,这样的专业身份又可以激励自己更加努力地实践从而提升领域的知识和技能(Foster & Shah, 2016)。因此,为了系统客观地评估编程初学者的计算思维能力水平和发展轨迹,我们需要深入分析协作编程过程中所展现的计算思维各能力维度,以及各维度下不同要素之间的关联。

3. 量化民族志

为了研究协作编程中学习者的计算思维发展过程,一种有效的方法是分析参与协作编程的学生之间的讨论交流内容。Leung (2012)认为,民族志研究有助于我们对社会性学习过程的理解。但是,传统的民族志研究是较为费时费力的质性分析,很难开展大规模的分析评估。为了解决分析海量会话数据的困难,Shaffer (2017)提出了一种叫做“量化民族志”的分析方法。量化民族志的关键环节是认知网络分析(Epistemic Network Analysis, ENA),这是一种对专业能力进行网络化建模的技术。ENA的核心思想是通过计算各个能力编码在对话的上下文中的共现次数,从而建立整个对话过程所体现

的、反映不同能力编码之间联系的网络化表征。它将专业能力的各个元素所构成的高维空间网络,通过奇异值分解的方法投影到二维平面上,各元素间线条的粗细代表元素间的关联强弱,亦即讨论中共现频次的高低。因此,基于学习讨论的内容,构建反映专业能力的认知网络图,可以分析并比较不同学生个体或学习小组的网络结构特点及差异。本研究试图将ENA应用于计算机编程教育中,分析学生的计算思维发展情况,采用类似一般新手-专家的比较研究方法揭示高低编程水平的学生其计算思维能力发展的特征、轨迹和差异。

综上所述,本研究主要探究以下两个问题:

问题1:在完成协作编程任务的过程中,高水平组和低水平组的学生在计算思维能力模式上各自具有怎样的特点(即计算思维的各能力维度之间呈现何种关联)?

问题2:在完成协作编程任务的不同阶段,高水平组和低水平组学生的计算思维能力的发展轨迹呈现怎样的特点?

三、模型建立

建立对计算思维的可操作评价模型,需要从评价维度与评价方法两方面着手。在评价维度上,本研究采用Brennan和Resnick的三维框架,将计算思维的评价维度界定为专业概念、专业实践和专业认识。此框架实现了对学生的计算思维的概念理解、从设计到开发的计算思维实践活动,以及学生对计算机科学的认知和观念转变这三个层面的关注与评估。在评价方法上,本研究采用“以证据为中心”的评价设计模式(Evidence-Centered Design, ECD)。该模式是Mislevy等人提出的关于能力评价的有效测评模式(Mislevy et al., 2003)。近几十年学习科学研究形成的共识是,深度学习的发生和能力的习得往往都是基于社会互动和情境化的。因此,能力评估需要在创设的实境学习情境之中,来获得关于行为的证据并对这些证据进行建模。而ECD正是这样一种设计、制作和提供教育评估的概念框架,它包括三个关键组成部分,分别是领域分析、领域建模、能力评估。领域分析是指关于特定领域的态度、能力以及专业知识的信息;领域建模是指

能够证明学生熟练程度的证据结构；能力评估是指能够分析学生熟练程度的评估工具或评估模型。在本模型中，笔者将依照该模式对计算思维的发展评估进行设计。具体而言，领域分析模块界定为计算思维领域的能力评价及发展轨迹分析；领域建模中的证据结构包含计算思维发展过程中的表现行为、话语、制品、过程及反馈，其中关于证据的获得笔者设计了基于项目的协作学习活动，以实现构建社会性的、真实问题的学习环境；能力评估模块依照计算思维的评价维度设计了对专业概念、专业实践、专业认识三个视角的编码框架。三个模块共同构成了协作编程学习中的计算思维评价模型（如图1所示）。

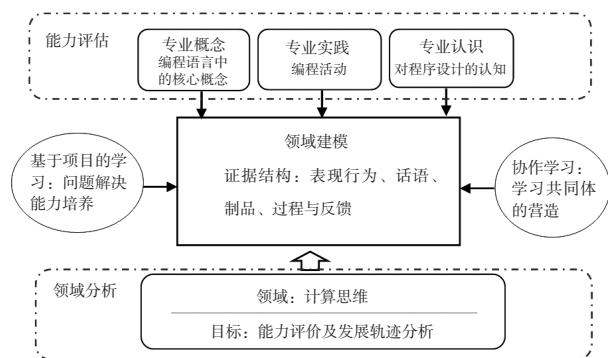


图1 协作编程中的计算思维评价模型

四、实验研究

1. 研究对象

在本研究中，笔者以华东地区某大学教育技术专业一年级的“C/C++程序设计基础”课程为研究情境。在该门课中有四周时间，学生将以小组形式（每组3~4名学生，学生自由组合）在课外共同完成两个相关联的游戏开发项目作为课程作业，学生在参与课程学习之前基本没有编程经验。项目要求各组学生在课后约定同一时间和地点共同完成项目任务。每次编程过程中，每组学生需共同使用一台电脑进行编程操作，其中一名组员负责编写代码，其他组员负责查看代码是否正确，并共同讨论编程思路。由于研究对象是编程初学者，对操作的不熟练使得实验过程中的多人协作活动更侧重于二人协作编程的模式，即一人编写代码，其余人提出建议。每一次的讨论，编程者的角色可以更换，整个项目分多次完成。本研究根据项目完成表现选择了

最具代表性的两组进行深度分析，并采用录屏的方式收集分析数据。

2. 任务设计

该课程中的编程项目需要学生利用所学的编程知识完成真实的应用程序开发，并随着学生所掌握编程技能和知识的增多，项目会逐渐加大难度。

图2是本研究选择的两个项目的主界面。第一个项目要求学生开发基于控制台的字符格式的鱼塘模拟应用程序。在这个程序的主界面中，鱼缸、鱼、鱼饵和鱼钩用不同的显示字符表示。鱼塘中所有物体按既定的方式运动。例如，随机出现的四条鱼在鱼塘中自左向右水平匀速运动，到右侧边缘后再重新从左侧出现，循环往复。第一个项目的学习目标是帮助学生掌握结构化程序设计的基本技能，以及基本的C语言语法知识，如循环、分支、数组、参数传递和函数调用等。第二个项目是在第一个项目基础上，实现图形用户界面的鱼塘模拟。完成项目的目的是希望学生初步建立面向对象的编程思想，掌握相关概念（如类和对象、类的属性和方法、对象之间的调用、类的继承）以及图形用户界面的编程技术（包括绘图、动画、交互事件等）。



图2 项目界面（左右分别为项目1和项目2的主界面）

3. 数据采集与分析

两组学生在协作编程过程中都需要进行录屏，除了提交编程作品之外，还要提供协作编程全程讨论交流的音视频数据。研究所采用的录屏软件是Camtasia，用于记录小组在Visual Studio 2015 IDE环境下的编程过程，以及编程过程中的组内对话，并通过笔记本电脑前置摄像头记录所有成员在编程中的交流情形。这样，通过摄像头录制的学生交流的视频信息可以帮助区分交流语音中不同学生的发言内容，也可以和录屏的代码编写过程进行相互印证。数据分析过程包括对各组在协作编程过程中的会话内容进行采集、转录、编码，最后采用认知网络建模的方法进行量化分析。具体的认知网络分析

方法详见第五部分。两个小组在完成两个项目的合作编程任务中生成的录屏数据时长473分钟,以每个组员的发言作为编码单位,将语音数据转录为文字稿,得到的发言条数为1533条。

4. 编码方案

笔者在Brennan和Resnick(2012)提出的三维计算思维框架的基础上设计编码方案,用来分析学生在协作编程中的对话数据(见表1)。第一个维度的专业概念是指编程语言中的核心概念,如顺序、循环、条件、运算符和数据结构等;第二个维度的专业实践是指学生的编程活动,如代码的增量和迭代、测试和调试、重用和混合、模块化和抽象化;第三个维度的专业认识是指学生对于程序设计的认识,比如将软件程序作为自我表达的工具,或者是对计算技术的功能和局限性提出疑问。

表1 计算思维的编码方案

维度	编码	定义
专业概念	顺序 (C.Seq)	可以由计算机顺序执行的步骤或指令序列
	循环 (C.Loo)	反复执行某段代码的一种计算机处理过程
	条件 (C.Con)	根据特定条件执行一些语句,用于支持多种情况的表达
	运算符 (C.Ope)	为数学、逻辑和字符串表达式提供支持,使程序员能够执行数字和逻辑运算,以及字符串操作
	数据 (C.Dat)	数据类型以及对数据的存储、检索和更新
专业实践	增量和迭代 (P.Inc&Ite)	识别编程规划和设计实施过程中的关键概念,是不断完善和迭代的过程,它将复杂任务分解为简单的子任务和目标
	测试和调试 (P.Tes&Deb)	解决程序的错误和问题,处理意料之外的编程结果
	重用和混合 (P.Reu&Rem)	寻找类似的想法和代码来构建程序,创造出比学生已有的代码成果更复杂的东西
	抽象化和模块化 (P.Abs&Mod)	识别共性的模式,并对具体实例进行归纳抽象
专业认识	表达 (I.Exp)	将计算视为自我表达和功能创新的媒介
	问题 (I.Que)	对计算技术的功能可供性和局限性提出疑问

为了对小组项目活动的会话内容进行编码,首先将音视频转录成文字,而转录稿中的对话记录体现了发言人的对话轮次。每轮发言的内容(即一条对话记录)作为一个分析单元,可以归类到一个或多个编码中。所有对话记录由两名研究人员独立编

码,编码结果具有很好的一致性(Kappa=0.83)。关于编码评分的差异,研究人员通过讨论协商解决。

五、分析结果

通过评估各组的程序作品来确定小组的编程表现,课程教师从四个维度对学生作品进行打分,包括代码正确性、功能完整性、代码简洁性和设计创意性。每个维度的分数范围是0~5分,满分为20分。对全班15个组的成绩进行统计后得出,平均分为14.50,标准差为4.19。将平均分作为分割线,将所有小组划分为高能力组和低能力组,从高能力组中各选择一个小组的学生,进行量化民族志分析。高能力组(总分为18.6)中,组员是三名男生,分别匿名用GZR、ZMZ和LYX表示;低能力组(总分为13.84)中,组员是两名男生和一名女生,分别匿名用WXL、ALM和CJM(女)表示。

为了说明高能力组和低能力组在协作编程活动中的交流互动模式,笔者选择其中一个会话片段作为定性的会话文本分析案例,以展示学生编程活动中的讨论内容,然后通过认知网络分析的结果来进一步说明所节选的会话片段如何反映两组学生的计算思维能力特点和发展趋势。

1. 在控制台上输出鱼塘

第一个项目的第一个任务是要求学生输出一个鱼塘,32列代表鱼塘的宽度,8排代表鱼塘的深度(见图2左)。高能力组学生采用自顶向下的方式来分析解决问题,从分析探讨问题的计算表达(专业认识),到如何定义变量和方法(专业实践),再到研究方法中的参数选择和变量作用域的相关知识(专业概念)(见表2)。首先,#1 GZR和#2 LYX确定了任务目标并将目标分解为两个计算表达的任务步骤,即鱼塘生成和鱼塘输出。随后,#3 ZMZ提出在编程中实现第一个任务可以用到的方法。然后他们开始讨论方法的定义和调用这种数据模块化的操作,进而来实现生成鱼塘的功能(#5~#7)。

#8 ZMZ进一步提出疑问,就具体代码编写中定义变量所要考虑的作用域问题展开讨论。

相反,在编程初期,低能力组的谈论内容始终围绕计算概念展开(见表3),例如,数组的维度和变量类型(#2~#4)、嵌套循环(#5~#12)、打

表2 高能力组的方法定义和变量范围

次序	话语文本	编码
#1 GZR	我们需要使用数组来定义鱼塘，并做出一个用水填充鱼塘的功能。	C.Dat, P.Abs&Mod, I.Exp
#2 LYX	需要先将水注入鱼塘，然后输出。	C.Seq, I.Exp
#3 ZMZ	那应该把这个函数放在哪里，在主函数里面还是外面？	C.Seq, P.Abs&Mod
#4 LYX	你可以在主函数外面定义这个函数，然后在里面调用它。	C.Seq, P.Abs&Mod
#5 GZR	那我们先写这个填充鱼塘的函数，我们要考虑它需要多少个参数。	P.Abs&Mod
#6 LYX	首先，数组应该是其中之一，然后水应该是 char 类型的数据，也是一个参数吧？	C.Dat, P.Abs&Mod
#7 GZR	那么我们需要多少个参数呢？	P.Abs&Mod
#8 ZMZ	我们能不能在这个函数里，而不在主函数里定义鱼塘数组？	C.Dat, P.Abs&Mod
#9 GZR	主函数也必须有一个数组。	C.Dat, P.Abs&Mod
#10 LYX	这个数组在全局范围内是必需的。	C.Dat
#11 GZR	因为你需要在主函数中输出。	P.Abs&Mod, I.Exp

表3 低能力组中计算表达和循环的实现

次序	话语文本	编码
#1 ALM	我们需要用水填充这个数组。	I.Exp
#2 WXL	是的，我们要定义一个数组。因为数组中包含字符，所以我们需要定义 char 类型数组，对吧？	C.Dat
#3 CJM	一个非常大的数组……	C.Dat
#4 ALM	鱼塘的大小是32*8，我们需要把它填满水。	C.Dat, I.Exp
#5 WXL	是的，我们先把水填满，然后在某个地方放入鱼。现在我们用for循环来填充水。	I.Exp, C.Loo
#6 ALM	你打算怎样用循环来填满这个鱼塘？	I.Exp, C.Loo
#7 CJM	两个嵌套的循环。	C.Loo
#8 WXL	用 printf 来输出它，像这样。我们用第一个 for 循环来打印行。	C.Loo
#9 CJM	第一个循环是行还是列？	C.Loo
#10 WXL	没有区别，只要能遍历到每行每列就好。现在的问题是我们的行和列都是数字，现在我们需要……	C.Loo, C.Dat
#11 ALM	这只是地址的变化，我们需要把字符填入这个地址。我建议我们继续给 i 和 j 分配水，对吧？	C.Dat, C.Ope
#12 WXL	不，我们需要在for循环中输出水。我们不会将它分配给 i 和 j，而是将 a[i][j] 分配给数组中的某个地址。水填满数组后，当 i 和 j 改变时，位置也相应的改变。	C.Dat, C.Ope, C.Loo

印输出语句（#8）、变量赋值（#10~#12）。因为对一些基础概念比较模糊，他们需要花费一定的时间来厘清嵌套循环中内外循环的作用、二维数组中两

个维度的索引和二维数组变量取值的含义。

2. 认知网络分析

ENA 的核心思想是通过计算各个编码在对话上下文中的共现次数，从而建立整个对话过程中所有编码之间联系的网络化表征。ENA 分析过程如图3所示，首先，根据小组成员会话中每个人的发言对数据进行分割，基于计算思维的编码框架对每条发言的内容进行编码。其次，根据任意两个编码在一个固定长度的对话线索（称为对话窗）中共现与否，生成编码共现的邻接矩阵。最后，通过在整个对话中滑动对话窗得到的所有邻接矩阵累加并按照矩阵的行展开为邻接向量，通过归一化操作和奇异值分解，生成一个二维投影平面的旋转矩阵，用来表征高维编码网络在二维的投影。

通过建立基于全部对话数据的量化分析模型，可以为上述定性分析结果提供统计意义上的证据支持，并进一步确定高水平组和低水平组的计算思维能力发展趋势。笔者根据学生的对话编码数据绘制了两个小组以及各个学生的认知网络图。图4中的圆点代表每个学生的计算思维认知网络图的质心，正方形代表各组成员的网络图质心的平均值（或可视为小组网络图的质心），黑框代表各组的质心位置在95%水平上的置信区间。

认知网络分析结果显示，生成的二维投影中第一维度（X轴）占数据总体方差的34%，第二维度（Y轴）占数据总体方差的27%。从图4可以看出，两个组的认知网络在第一维度的数据差异是显著的（高能力组 $M=0.24$ ，低能力组 $M=-0.24$ ， $t=4.143$ ， $p=0.019<0.05$ ，Cohen's $d=4.143$ ），而在第二维度的数据却没有显著差异。这两个二维投影的坐标维度的含义可以通过图3的认知网络结构图进行解释。

为了进一步分析这两个组认知网络结构之间的差异，笔者绘制了这两个小组整体上的平均认知网络（如图5）。ENA 网络模型中线条的粗细和饱和度代表元素间连接的强弱。图5b显示，高水平组各元素间的连接更为复杂，在测试和调试以及与循



图3 认知网络分析过程

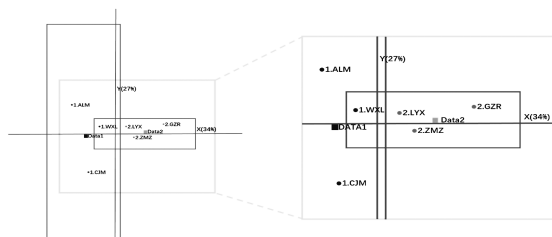


图4 投影空间中高能力组（浅灰）和低能力组（深灰）学生的计算思维认知网络（右图为左图中心区域的局部放大）

环之间的连接相较低能力组更强，说明高能力组学生在完成编程任务时更多关注代码的测试和调试循环（C.Loo）之间的关系；而低能力组更多是在提出计算思维相关的疑问、重用和混合、增量和迭代以及序列之间进行交流。此外，从图5中还可以看出，与其他元素连接最紧密的是计算表达，两个组都强调了其他元素和计算表达之间的联系。通过将图5a和图5b相减可以得出图5c，图5c能够更清晰地看出高低能力组学生平均认知网络的差异。如果两组存在元素间连线的重叠，最终呈现出来的是连接较强组的线条，并且线条粗细都会相互叠加。从图5c可以看出，低能力组更侧重于左边区域元素间的连接，高能力组更侧重于右边元素的连接，这也印证了前文得出的二者在第一维度上差异性显著、在第二维度上无显著性差异的结论。

接下来，笔者通过分析这两个组在完成各项项目不同阶段的计算思维认知网络轨迹，以期了解高低能力组的计算思维变化趋势。由于高能力组分3次完成了第一个项目，1次完成了第二个项目，而低能力组分3次完成了第一个项目，通过2次讨论后完成了第二个项目，因此可以对各组每次的谈话分别构建认知网络图，从而得到高能力组学生的4次计算思

维网络图和低能力组学生的5次计算思维网络图。

图6中各点是根据每个学生的各次讨论生成的网络图质心，可以发现，两组学生的计算思维能力呈现出不同的发展轨迹。在项目开始时，不同小组学生的计算思维各维度呈现显著差异（具体表现为每位学生的认知网络图质心在二维投影平面上所处的位置），而组内学生之间的网络图质心位置相对接近；但值得注意的是，随着协作编程过程的深入，两组成员的认知网络图质心位置趋于一致。图6显示，虽然高能力组学生在开始时更多地关注其他计算思维的能力维度与抽象化和模块化、以及程序测试之间的关系，而低能力组更多地关注其他能力维度与重用和混合之间的关系，但网络轨迹显示这两

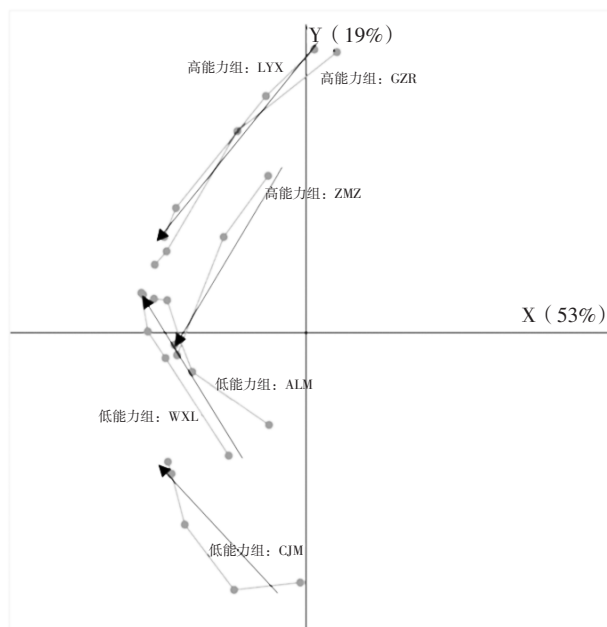
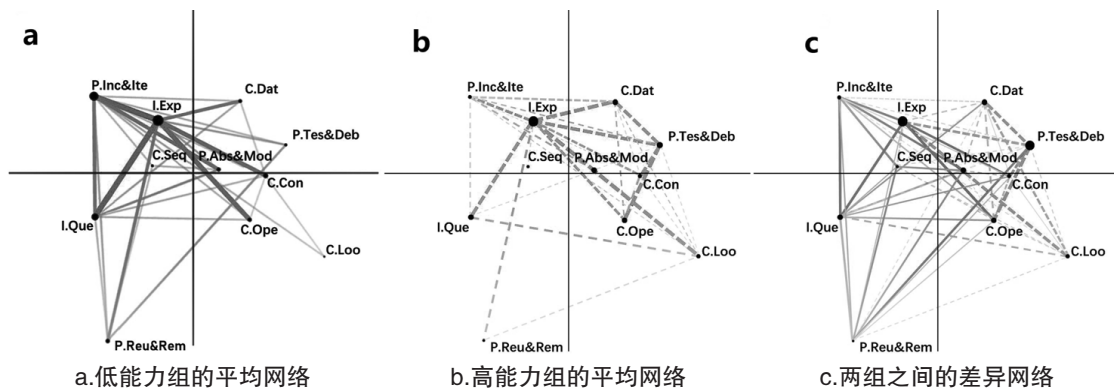


图6 投影空间中两组6名成员的网络模型质心轨迹（箭头表示从第一次协作编程讨论到最后一次讨论的网络图质心移动方向）



a.低能力组的平均网络

b.高能力组的平均网络

c.两组之间的差异网络

图5 各组的平均认知网络（虚线代表高能力组的连线强度大于低能力组，反之为实线连线）

组最终趋近于相似的计算思维能力模式，即均建立起其他能力维度与增量和迭代之间的显著联系。

六、研究讨论

1. 计算思维结构存在双向思维特征

在协作编程的不同阶段，笔者选取两组学生的典型会话片段进行定性分析，对具体问题解决过程中所体现的计算思维能力特征进行了微观分析和诠释。此前Chao (2016)的研究揭示了编程实践、程序设计思维以及编程表现之间的关系，本研究的质性分析进一步拓展了他的研究成果，可以看到编程问题解决存在自顶向下和自底向上的双向思维过程。从自顶向下的角度来看，学生首先应该分析需求，思考如何通过程序设计表征问题，然后进一步分析实现这一设计目标使用的具体技术方案。而从自底向上的角度来看，学生还需要考虑技术实现的可能性和局限性，从而寻找更优的解决方案并对原型设计进行改进，甚至从根本上重构整个系统实现的方式。高能力组和低能力组就具体编程任务的讨论，体现了两组学生的计算思维结构特征分别侧重于建立专业认识和专业实践，以及专业概念和专业实践之间联系的差别。

2. 高低能力组学生的计算思维发展存在趋同演化趋势

通过对各组会话内容进行编码和建立认知网络模型，可以从定量角度进一步揭示高低水平小组的计算思维能力差异和发展轨迹的不同。追踪分析发现，虽然在项目最初不同水平的编程小组之间，其计算思维特征存在显著差异，但随着项目的不断迭代，低能力组表现出的思维特征不断向高水平组趋同。具体而言，在项目一开始高能力组更多关注代码功能模块，同时在协作编程期间他们不断地提高调试技巧；而低能力组在协作编程过程中更多关注代码行的次序正确与否，并且在编程过程中主要采用复制和粘贴代码的方式来完善程序，是一种相对低效、浅层次的程序修补策略。随后高能力组的计算思维网络从抽象设计和代码的模块化，延伸到创建和应用不同类型的数据以实现其设计，最终不断完善和改进程序。然而，在低能力组的编程过程中，一开始他们试图重用和混合已有的代码片段，

这表明他们没有在编程之前制定计划，并没有对项目问题和编写的代码进行深入思考，他们甚至质疑技术实现的可行性；但随着编程经验的累积和对已完成的程序重新审视和反思，在后续程序开发中，他们逐步意识到不断迭代优化程序实现方式的重要性，并且加深了对面向对象设计思想的理解，最终他们的思维结构特征逐步向高水平组靠拢。

综上所述，已有研究表明定量的民族志方法是一种很有前景的对质性数据进行量化分析的方法，它可以推广至其他专业领域的学科思维或能力评价中，并为协作学习过程的分析评价提供了新的思路。在研究学生的专业能力发展时，我们首先基于理论驱动的认知模型设计编码方案，再将会话数据进行编码，利用认知网络分析工具以及跟踪分析的方法分析小组合作编程在项目编程初期、中期和末期的能力模型特征和轨迹，从而建立学生的领域认知模型。这就实现了将社会性学习过程中的会话交流外显化，使我们可以获得基于学习证据的个人及小组的能力水平和发展轨迹，并据此提供更有针对性的协作学习干预。当然，本研究尚属探索性质，还存在一些不足。例如，我们只选择了两个具有代表性的小组的编程项目展开分析，由于样本量较小，使得本研究建立的计算思维模型未必可以直接推广到其他程序设计课程情境和学生群体中。未来的研究中将增加个案，提高研究结果的泛化效果。此外，在研究过程中我们主要将会话编码为计算思维的特定能力维度，未采用其他已采集的数据，且只分析了会话中是否涉及到该维度能力，而并未评价这个维度上能力的高低。未来研究中我们可以进一步改进认知网络分析法，采用多模态数据建模方法以及加权网络来实现对学生专业认知能力更为精细的建模分析。

参考文献：

- [1]Beck, L., & Chizhik, A. (2013). Cooperative Learning Instructional Methods for CS1: Design, Implementation, and Evaluation [J]. ACM Transactions on Computing Education, 13(3): 10–21.
- [2]Berland, M., & Lee, V. R. (2011). Collaborative Strategic Board Games as a Site for Distributed Computational Thinking [J]. International Journal of Game-Based Learning, 1(2): 65–81.
- [3]Brennan, K., & Resnick, M. (2012). New Frameworks for

Studying and Assessing the Development of Computational Thinking [C]// Paper presented at the Proceedings of the 2012 Annual Meeting of the American Educational Research Association. Vancouver, Canada.

[4]Chao, P. Y. (2016). Exploring Students' Computational Practice, Design and Performance of Problem-Solving Through a Visual Programming Environment[J]. Computers & Education, 95: 202-215.

[5]Denner, J., Werner, L., & Campe, S. et al. (2014). Pair Programming: Under What Conditions Is It Advantageous for Middle School Students?[J]. Journal of Research on Technology in Education, 46(3): 277-296.

[6]Foster, A., & Shah, M. (2016). Examining Game Design Features for Identity Exploration and Change[J]. Journal of Computers in Mathematics and Science Teaching, 35(4): 369-384.

[7]Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field[J]. Educational Researcher, 42(1): 38-43.

[8]Kafai, Y. B., Lee, E., & Searle, K. et al. (2014). A Crafts-oriented Approach to Computing in High School: Introducing Computational Concepts, Practices, and Perspectives with Electronic Textiles[J]. ACM Transactions on Computing Education, 14(1): 1-20.

[9]Kalaian, S. A., & Kasim, R. M. (2014). Small-Group vs. Competitive Learning in Computer Science Classrooms: A Meta-Analytic Review. In Queiros, R. (Ed.). Innovative Teaching Strategies and New Learning Paradigms in Computer Programming[M]. IGI Global: 46-63.

[10]Khosa, D. K., & Volet, S. E. (2014). Productive Group Engagement in Cognitive Activity and Metacognitive Regulation During Collaborative Learning: Can It Explain Differences in Students' Conceptual Understanding?[J]. Metacognition and Learning, 9(3): 287-307.

[11]Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A Validity and Reliability Study of the Computational Thinking Scales (CTS)[J]. Computers in Human Behavior, 72: 558-569.

[12]Kwon, K., Liu, Y. H., & Johnson, L. P. (2014). Group Regulation and Social-Emotional Interactions Observed in Computer Supported Collaborative Learning: Comparison Between Good vs.

Poor Collaborators[J]. Computers & Education, 78: 185-200.

[13]Leung, W. C. (2002). Why Is Evidence from Ethnographic and Discourse Research Needed in Medical Education: The Case of Problem-based Learning[J]. Medical Teacher, 24(2): 169-172.

[14]Lye, S. Y., & Koh, J. H. L. (2014). Review on Teaching and Learning of Computational Thinking Through Programming: What Is next for K-12?[J]. Computers in Human Behavior, 41: 51-61.

[15]Mangalaraj, G., Nerur, S., & Mahapatra, R. et al. (2014). Distributed Cognition in Software Design: An Experimental Investigation of the Role of Design Patterns and Collaboration[J]. MIS Quarterly, 38(1): 249-274.

[16]Mislevy, R. J., Almond, R. G., & Lukas, J. F. (2003). A Brief Introduction to Evidence-Centered Design[R]. ETS Research Report Series.

[17]Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which Cognitive Abilities Underlie Computational Thinking? Criterion Validity of the Computational Thinking Test[J]. Computers in Human Behavior, 72: 678-691.

[18]Shaffer, D. W. (2006). Epistemic Frames for Epistemic Games[J]. Computers and Education, 46(3): 223-234.

[19]Shaffer, D. W. (2017). Quantitative Ethnography[M]. Madison, Wisconsin: Cathcart Press.

[20]Wang, S. L., & Hwang, G. J. (2012). The Role of Collective Efficacy, Cognitive Quality, and Task Cohesion in Computer-Supported Collaborative Learning (CSCl) [J]. Computers & Education, 58(1): 679-687.

[21]Williams, L., & Kessler, R. (2002). Pair Programming Illuminated[M]. Addison-Wesley Longman Publishing Co., Inc..

[22]Zhong, B., Wang, Q., & Chen, J. et al. (2016). An Exploration of Three-dimensional Integrated Assessment for Computational Thinking[J]. Journal of Educational Computing Research, 53(4): 562-590.

收稿日期 2018-11-11

责任编辑 刘选

(下转第94页)

The Study of Distance Learners' Emotion States and Its Distinguishing Characteristics in Online Learning

ZHAO Hong, ZHANG Xinmiao

Abstract: Emotion is the attitude towards objective things, which influences learning quality and efficiency. Because of the time-space separation in online learning environment, emotional concern is very important for learners. The previous studies in relation to adult learners' emotion states focused on the methods and strategies of emotional management. Understanding learners' emotion states and its changing track in online learning environment is vital to analyze the students' studies and provide personal learning support. Based on the analysis of emotional states and its changing track, this study found that: (1) Learners were mainly in positive emotion throughout the whole learning process. However, the emotion intensity changed overtime, showing that positive emotion was on the decline, while the negative emotion was just the opposite. (2) The significant difference was found between male and female learners with the learning going on. The emotional states of male and female learners showed opposite changing tracks. (3) There was a significant difference among learners with different educational levels. (4) The emotional states of learners with different online learning experiences were not significantly different.

Keywords: Distance Learners; Online Learning; Emotion States; Variance Analysis

(上接第84页)

The Development Trajectory of Computational Thinking in Cooperative Programming: A Quantitative Ethnography Approach

WU Bian, WANG Ge

Abstract: With the rapid development of information technology in the 21st century, computational thinking is deemed as an important skill in many professions. Research in the field of computer education found that collaborative programming can better help students to learn programming, but in the process of collaboration, the dynamic development of students' computational thinking is difficult to measure. Quantifying ethnography is not only a method for quantitative analysis of qualitative data, but also a technology for network modeling of professional competence, which provides a new idea for the analysis and evaluation of collaborative learning process. Applying the method to the analysis of conversational communication between group members in the process of collaborative programming activities is conducive to constructing a cognitive network model of the relationship between different dimensions of students' computational thinking ability, and revealing the level of computational thinking ability of groups with different ability levels and development trajectories. The empirical study of the "C/C++ Programming Foundation" course in the first year of undergraduate study found that the computational thinking network structures of high-level and low-level students are significantly different at the beginning, but with the continuity of collaborative task, the thinking patterns of the two groups of students tend to be similar. The results of this study show that collaborative programming can promote the improvement of beginners' computational thinking ability, and the quantitative ethnography can provide a large-scale assessment method based on qualitative data for the evaluation of computational thinking ability.

Keywords: Computational Thinking; Collaborative Programming; Quantitative Ethnography; Epistemic Network Analysis